

- Make it easy to report issues.
 - Have a semi-structured form to allow the issue to be triaged
- Automated code review (e.g. “codechecker”) in plugin DB/cibot (MDLSITE-2857 ?)
- Share Partner code review process/standards/template
 - Structured review template
- Maybe allow Moodle sites to assert that they are a production/development site
- Who performs reviews and who is “trusted” that their review is believable.
 - Certified Reviewer role?
 - Badges associated with plugins.
 - Editorial level review
- 2 kinds review
 - Did I like it?
 - Is it implemented well?
- Poorer compliance with coding standards may correlate with poorer quality
- Featured developer
- Issue badges
- Use combination of factors to determine a ranking rather than just latest (expand MDLSITE-2540)
- Some stats
 - Existence of tests may be an indicator
 - Unit Test
 - Behat test
 - Compliance with codechecker
- Meta-reviews
 - Was this review useful / not useful
- “Report this plugin” mechanism in Plugins DB
 - Moodle HQ could maybe make a definitive statement about goodness/badness
- Sandboxing plugins?
- Use of tags to highlight traits of plugins? ([MDLSITE-1428](#))
 - e.g. “remotelearner plugin”, “securityrisk”
- Display development stats (activity, updates, #issues raised)
 - diff stats for each version
- Reviews tend to be point in time and associated with specific version
- Certified Developers? or highlighting contributions to core
- Tie developer information displayed on plugin DB to Moodle.org profiles
- Is there a way to push an update without having to do anything in the Moodle Plugin DB. -from Sam Marshall (not sure I got this down correctly-MH)
- Should certain people be allowed to upload plugins and version without them being checked.
 - Currently 1st version reviewed, subsequent versions not reviewed
 - Persistent offenders can be black listed
- Should developers gain or lose trusted status?
- Allow people to see unapproved plugins and install them if they chose to.
 - Filters to display reviewed, non-reviewed, trusted, not trusted (etc) plugins
- Currently can only install plugins when marked as compatible with a version
 - An “I’m feeling crazy” button that allows a plugin to be installed on a non-released Moodle Version -linked to certification quiz :-)
 - workaround: said crazy person can also tweak the beginning of the url in those install buttons. other versions installs were left out of interface to reduce fails but ‘crazy’ should be warning enough.
- Mechanisms to manage gaming of updates and downloads

- An update pushes the plugin to the top of the latest plugin
- Additional taxonomies ([MDLSITE-1428](#))
 - Current categories are basically “api”s so really only meaningful developers
 - Who is the plugin for?
- Curated lists of plugins
 - somewhat like <https://moodle.org/plugins/browse.php?list=award&id=1> perhaps?
 - We will need curators. Simply, we can modify and open up the awarding system to others for their own list OR create ‘favouriting plugins’ mentioned below.
- Hook a reporting/reviewing mechanism into the Moodle manage plugins list (from within a site) ([MDLSITE-3006](#))
 - Report issue option from within the plugin list
- Favoriting plugins
 - Display number of users that the plugin is favoured by/on a curated list
- On uninstallation provide a mechanism to make a review/report to the plugin db. ([MDLSITE-3006](#))
- A “playground” site containing important plugins.
- Is it easy to make a youtube video to demonstrate a plugin?
 - note: plugin description area allows youtube embedding after video is created.
- Plugins site:
 - Not necessary:
 - 2 search boxes not necessary
 - Version at top of the screen
 - Partner advert does take up space
 - Whitespace usage?
 - Get rid of categories?
 - See Firefox add-ons?
- Put additional information about plugin in the version.php ([MDLSITE-3006](#))
 - Bug reporting URL
 - Buttons generated from some of this information?